Problem Statement

Given a set of text documents (without any initial internal hierarchy) find out the pairs (or groups) of documents that are significantly similar to one another and suggest possible plagiarism.

Basics

- Building on the task given before, one can use Levenshtein Distance as a simple measure of similarity. This can be a simple yet effective measure for small document sizes. Other measures of edit distance can also be used.
- Cosine Similarity measure and can be used as a simple metric to measure similarity between two documents. This involves transforming the file into a vector and using similarity measures in that vector space (like the cosine of angle between vectors using inner products).
- A slightly more advanced method involves the use of data structures called suffix trees. This can be used to match exact substrings like sentences very quickly by preprocessing data. This can be extremely effective in detecting copy & pasting.

TASK 1

Choose one of these models (or even a hybrid version of some of these) that most suits your problem and build a basic working code that achieves your goal.

PROJECT SPECIFIC METHODS

- One of the main methods for improvements for text documents is Natural Language Processing. This involves exploiting the fact that they are written in a particular language (which we will assume to be English). For example, one can throw away frequently used words like “the”, “is”, etc, thus reducing time and space needed for processing. This can help detect plagiarism arising from changing some words in the document without changing the meaning.